Welcome 1

Welcome to THINK C or Symantec C++. This is the User's Guide for both. This chapter tells you what's in your THINK C or Symantec C++ package, what equipment you need, and what you need to know to write programs on your Macintosh.

If you don't read manuals

Read this chapter, the next chapter, and one of the tutorials. Be sure to keep this manual nearby to answer any questions that come up as you work. If you are using the Symantec C++ compiler, be sure to read the "Welcome" section in the *Symantec C++ Compiler Guide.*

If you're an experienced THINK C user

Read Chapter 2, "Installation," before installing the new version. If you read nothing else, read Appendix A, "What's New," to learn about the new features in THINK C and Symantec C++.

If you're new to THINK C or the C language

Read Chapter 3, "Tutorial: Hello World," to learn how to run simple programs in THINK C. If you need a book on C, see "Learning C" later in this chapter. If you're learning C from a book written for Unix or MS-DOS computers or if you have experience working with C in another operating system, look at the sections "Standard Libraries Intro" and "Console Package Intro" in the THINK Reference *Standard Libraries Reference* database. Read the section, "Using the Online Standard Libraries Reference" in Chapter 21, "Using the Standard Libraries," to learn how to use the online databases.

Contents

The THINK Development Environment. 5

What You Need 6
Which Macintosh models? 6
How much disk space? 6
Which System/Finder? 6

1 Welcome

What's in the Package 6

What's in the User's Guide 6
- Getting Started 7
- Learning the THINK Environment 7
- Using THINK Project Manager 7
- Using the THINK C Compiler 9
- Using Object Extensions 11
- Advanced Topics 11
- Reference 11
- Appendixes 12

Conventions 12

What You Should Know 13
- Learning C 13
- Learning to write Macintosh programs 15
- Apple Computer, Inc.. 17
- Apple Programmer's and Developer's Association (APDA) . 18
- CompuServe 18

The THINK Development Environment

THINK C and Symantec C++ provide a unique development environment for the Macintosh. It features very fast compilers, a faster linker, a time-saving optimizer, an integrated debugger, a text editor, an auto-make facility, and a project organizer that holds all the pieces together. And if you have Symantec C++, it also has a user interface generator that helps you write your code. Because the editor, the compilers, and the linker are all components of the same application, the THINK Project Manager knows when edited source files need to be recompiled. And if you edit a header file, the auto-make facility recompiles all the source files that depend on it for declarations.

With THINK C and Symantec C++, you can build Macintosh applications, desk accessories, device drivers, and any kind of code resource. The standard C libraries include all the functions specified in the ANSI C standard, as well as some additional Unix operating system functions. Symantec C++ includes an extendable IOStreams library, as well as a library for operations on complex numbers.

You can run your program from the THINK environment as you work on it. Your program runs exactly as if you had opened it from the Finder, not under a simulated environment. And if you use System 7 (or System 6 with MultiFinder), your program runs in its own partition while the THINK Project Manager remains active, so you can examine and edit your source files as you watch your program run.

The THINK development environment includes a source-level debugger that lets you debug your code exactly as you wrote it. You no longer need to translate assembly language back into C. The debugger lets you set breakpoints, step through your code, debug objects, examine variables, and change their values while your program is running. And because the debugger works along with the THINK Project Manager, you can edit your source files while you're debugging.

Symantec C++ includes a user interface generator called the Visual Architect. It lets you graphically create your application's windows and dialogs, and then it writes C++ code to implement them. Later you can modify the code to handle your application's specific needs.

What You Need

THINK C requires that you have a hard drive and at least 4 megabytes (MB) of RAM (8MB is recommended). Symantec C++ requires that you have a hard drive and at least 8MB of RAM (12MB is recommended). They both use temporary memory when it is available and run in real or virtual memory in both 24- and 32-bit modes.

Which Macintosh models?

You can run THINK C and Symantec C++ on the Macintosh Classic, LC, Plus, Portable, SE series, II series, Performa series, PowerBook series, Centris series, and Quadra series.

How much disk space?

The complete THINK C system takes up about 11.1MB on your disk, not including your own files. The complete Symantec C++ system takes up about 17MB. The actual size of your system may be smaller, depending on the kinds of programs you work on. You can customize your installation to use less disk space.

Which System/Finder?

Use the latest System and Finder provided by Apple. THINK C and Symantec C++ require at least System 6.0.5. The THINK Project Manager is designed to work best under System 7 or System 6 with MultiFinder.

What's in the Package

The THINK C package consists of five 1.4MB diskettes, this User's Guide and the *THINK Class Library Guide*. Symantec C++ contains eight 1.4MB diskettes, this User's Guide, the *Visual Architect and THINK Class Library Guide*, and the *Symantec C++ Compiler Guide.*

What's in the User's Guide

This guide is organized in eight parts: Getting Started, Learning the THINK Environment, Using THINK Project Manager, Using the THINK C Compiler, Using Object Extensions, Advanced Topics, Reference, and Appendixes. Each chapter begins with an introduction that describes what's in the chapter, followed by a table of contents for the chapter.

Getting Started

This is the part you're reading. It contains this chapter and the installation instructions. Even if you don't read manuals, be sure to read the installation instructions in Chapter 2.

Learning the THINK Environment

This part contains the following four tutorials.

This chapter	Describes...
Tutorial: Hello World	How to write a small program that uses the standard C libraries. It introduces you to the basics of using the THINK environment.
Tutorial: MiniEdit	How to build a Macintosh application. This tutorial is based on the Sample program in *Inside Macintosh*. It shows you how to fix bugs, how to use resource files, and how to build a double-clickable application.
Tutorial: Bullseye	How to use the THINK debugger.
Tutorial: LearnOOP	How to use the THINK C object extensions.

Using THINK Project Manager

This part contains eight chapters that describe how to use the THINK Project Manager to develop your applications.

This chapter	Describes...
Overview	How to write a program using the THINK Project Manager. The rest of the chapters in this part of the manual discuss specific aspects of the

	components of the THINK development environment.
Organizing Projects	How to create and work with projects. Projects are at the heart of the THINK Project Manager. A project keeps track of all your source files and stores information needed to debug, compile, and create your program.
Organizing Files and Folders	Organizing your files and folders correctly helps the THINK Project Manager work faster. This chapter tells you how to set up your files and folders, and how to move your files from one folder or machine to another.
Editing Files	How to use the built-in THINK editor. The editor has some features, including powerful search routines, that make editing your source and header files easier.
Running Programs	How to run a program from the THINK Project Manager. The chapter also explains the following: how to compile files without running your program, how to examine what the compiler does with your code, how to pre-compile headers so your projects compile more quickly, and what happens when the THINK

	Project Manager can't run your program.
Debugging Programs	How to use the Project Manager's source level debugger, which lets you debug your application the way you wrote it, in source code.
Building Projects	How to build different types of projects such as applications, code resources, and libraries.
Browsing Classes	How to use the Class Browser. The class browser is a tool that displays all the classes defined in your project as a tree chart.

Using the THINK C Compiler

This part contains seven chapters that explain how to use the THINK C Compiler to compile code for your projects.

This chapter	**Describes...**
Writing C Code	How the THINK C compiler implements some features of the C language and how it adds to the language.
Compiling C Code	How to compile ANSI-conformant code, how to choose the precision with which THINK C type-checks your code, and how to produce programs that are tuned for a specific processor.

Optimizing Code	How optimizers work and how to use the THINK C optimizer.
Calling Toolbox Routines	How to call the Macintosh Toolbox routines from your programs. It also explains how to write Pascal callback routines and how to use some THINK C compiler extensions that help you write programs for the Macintosh.
Porting Code	How some compiler options can help you port code from other compilers.
Writing Inline Assembly	How to use assembly language in your THINK C programs with the inline assembler. This chapter also explains C and Pascal calling conventions so your assembly language routines will integrate smoothly with both your C functions and with the Macintosh Toolbox routines.
Using the Standard Libraries	The standard libraries that come with THINK C and Symantec C++, and how to use them with your project. The chapter also explains how to modify them for your own purposes and how to use THINK Reference.

Using Object Extensions

This part contains two chapters describing the basics of object-oriented programming and how to use the THINK C compiler object-oriented extensions to the C language.

This chapter	Describes...
Object-Oriented Programming	The basics of object-oriented programming (OOP). This chapter uses examples and comparisons to procedural programming.
Using Objects in THINK C	How to use the THINK C compiler's object-oriented extensions to the C language.

Advanced Topics

This part contains three chapters dealing with advanced topics.

This chapter	Describes
Using SourceServer	How to use SourceServer, a source code version control system.
Profiling Projects	How to use the profiler and profiler++ libraries that come with the THINK Project Manager.
Using Object Files	How to use the integrated MPW .o translator.

Reference

This part contains five reference chapters.

This chapter	Describes...
Using THINK Rez	How to write text files that compile with THINK Rez to produce resource files.

Using SADeRez	How to use SADeRez to create resource description files from resource files.
Language Reference	THINK C's implementation of the C language. Its sections are keyed to the ANSI standard.
THINK Project Manager Menus	The THINK Project Manager menu commands.
Debugger Menus	The THINK Debugger's menu commands.

Appendixes

This part contains the four following appendixes.

This chapter	**Describes...**
What's New	The new features in THINK C and Symantec C++ for Macintosh.
Troubleshooting	How to find solutions for common problems.
Error Messages	The error messages that the THINK Project Manager and compilers generate. Many descriptions contain examples and tips on solving problems.
Glossary	Definitions for many of the terms used in this manual.

Conventions

Names of menus, commands, and dialog boxes are in **boldface**. Book titles appear in *italic*.

Names of files, code fragments, resource names, function names, and variables appear in `typewriter face`. Metanames appear in *italic*.

All numbers are decimal. Hexadecimal numbers are written in C notation (`0x3EFA`) instead of Pascal notation (`$3EFA`).

In this manual, the term Toolbox routine means any routine described in *Inside Macintosh*. The Macintosh ROM actually consists of two different kinds of routines: operating system routines and Toolbox routines. Operating system routines deal with low-level aspects of the machine such as the file manager, the event posting mechanism, interrupts, and device management. The Toolbox deals with high-level aspects like the drawing environment, the window mechanism, menus, dialogs, and so on.

What You Should Know

This manual assumes you already know, or are at least learning, how to program in C or C++. If you're just getting started with C or C++, THINK C and Symantec C++ are great platforms.

If you're planning to write Macintosh applications, you should be familiar with the Macintosh Toolbox as described in *Inside Macintosh*, the official reference that describes the more than one thousand Macintosh Toolbox routines. The Toolbox is the set of operating system and user interface routines that make a Macintosh a Macintosh. It's beyond the scope of this manual to show you how the different parts of the Toolbox work together.

Learning C

As the popularity of C grows, more and more introductory-level books are appearing on the shelves. Some books assume that you're just learning how to program; others assume that you already know how to program in another language. Some books tell you how to use the development environment: the editor, the linker, the make facility. These things are done very differently in THINK C. So when you choose a book, choose one that doesn't dwell too much on these aspects of programming.

If you're learning C from a book, or if you're using THINK C to do course work, be sure to complete the first tutorial. It shows you how to set things up to write and run C programs that use the standard C libraries.

The standard references for the C programming language are Kernighan & Ritchie's *The C Programming Language, Second Edition* (Prentice Hall) and Harbison & Steele's *C: A Reference Manual* (Prentice Hall). *The C Programming Language, Second Edition* is an update to *The C Programming Language* that incorporates what was then the latest draft of the ANSI standard. These books assume that you're already an experienced programmer.

Standard C (Microsoft Press) by P. J. Plauger and Jim Brodie is a guide to writing C programs that conform to the ANSI C standard. Both of the authors were officers of the committee that drafted the ANSI standard.

Software Engineering in C (Springer-Verlag) by Peter Darnell and Philip Margolis is an excellent introduction to the C programming language. This book is ideal for new C programmers who have programmed in other languages.

C Traps and Pitfalls (Addison-Wesley) by Andrew Koenig is a good book for intermediate and advanced C programmers. It contains a detailed discussion of common C programming problems.

Numerical Recipes in C (Cambridge University Press) by Press, Flannery, Teukolsky, and Vetterling is a detailed technical description of numerical methods with implementation examples in C.

Portability and the C Language (Hayden Books) by Rex Jaeschke is a good book on writing portable C programs for advanced C programmers. It gives guidelines on writing programs that can be compiled with different compilers and run on multiple platforms. It points out changes introduced with ANSI C. You'll find it useful if you port code from one platform to another or from an old (pre-ANSI) version of THINK C to THINK C 6.0.

Portable C Software (Prentice Hall) by Mark R. Horton is also about writing portable C programs, but emphasizes porting among C compilers for Unix and MS-DOS.

The document describing the ANSI standard is *American National Standard for Information Systems—Programming Language—C*

X3.159–1989. Cost is $50, plus $5 for shipping and handling. To order a copy, write or call:

American National Standards Institute (ANSI)
Sales Dept.
1430 Broadway
New York, NY 10018
(212) 642-4900

Learning to write Macintosh programs

If you're new to programming the Macintosh, you might be overwhelmed by the complexity of the Macintosh Toolbox and unfamiliar programming techniques. When the Macintosh was introduced in 1984, commercial developers had a hard time figuring out how to get things to work correctly and very little was available for the casual developer.

The Macintosh is more complex today than it was in 1984, but now there are more sources of information. There are several good books that introduce programming the Macintosh and teach some of the finer points of using the Macintosh Toolbox. No matter which book(s) you choose to get started, *Inside Macintosh* is indispensable.

Inside Macintosh Volumes I-VI (Addison-Wesley) is the official reference that describes the more than 1,000 Macintosh Toolbox routines. At six volumes, it represents a hefty investment. The first three volumes cover the fundamentals. Volumes IV and V cover the additions and changes made with the introduction of the Mac Plus, Macintosh SE, and Macintosh II. Volume VI covers the changes introduced with System 7. If you're planning to write serious applications, you can't do without it.

Apple is in the process of publishing the *New Inside Macintosh* series, which will eventually replace *Inside Macintosh I-VI*. The *New Inside Macintosh* is rewritten to be easier to understand and up to date. It will contain all the information in *Inside Macintosh I-VI*, plus the information in the Technical Notes and a great deal of new information. To make it easier to use and update, each volume describes one part of the Macintosh Toolbox. Among the volumes now available in *New Inside Macintosh* are: *Inside Macintosh: Files*, *Inside Macintosh: Memory*, *Inside Macintosh: Macintosh Toolbox Essentials*, *Inside Macintosh: Text*, and *Inside Macintosh Quicktime*.

When this manual refers to sections in *Inside Macintosh*, it refers to the *New Inside Macintosh* whenever possible.

Note

Apple has not yet published all the books in the *New Inside Macintosh* series and recommends that you keep the original *Inside Macintosh I-VI* until it does so.

In addition to *Inside Macintosh*, Apple also publishes these books through Addison-Wesley:

- *Macintosh Human Interface Guidelines*
- *Technical Introduction to the Macintosh Family*
- *Programmers Introduction to the Macintosh Family*
- *Guide to the Macintosh Family Hardware, Second Edition*
- *Apple Numerics Manual, Second Edition*
- *LaserWriter Reference*
- *Inside AppleTalk*
- *Designing Cards and Drivers for the Macintosh Family, Second Edition*

Some books, like *Macintosh Human Interface Guidelines*, are useful for all Macintosh programmers. Others, like *Inside AppleTalk*, are meant for programmers working on specific kinds of applications. These books are available from APDA (see "Apple Programmer's and Developer's Association," later in this chapter), technical bookstores and computer stores, and some general bookstores.

THINK Reference, published by Symantec, is a hypertext program that gives you instant access to the critical system information that you need to program the Macintosh. It describes nearly all of the Macintosh Toolbox routines from *Inside Macintosh I-VI*, including information from the technical notes, code examples, and Symantec programmer tips. The information is cross-referenced, so you can look up related Toolbox routines quickly. It includes customizable bookmarks and templates that let you copy Toolbox calls right into your own program.

The *Macintosh C Programming Primer: Inside the Toolbox Using THINK C, Volumes I and II* (Addison-Wesley) is a good introduction to Macintosh programming for those already familiar with C. Volume I by Dave Mark and Cartwright Reed explains how to use the Toolbox, handle resources, write a Macintosh application; it also

gives examples that use some of the newer parts of the Macintosh system, such as the Notification Manager and HyperCard XCMDs and XFCNs. Volume II by Dave Mark covers more advanced topics, such as, Color QuickDraw and code resources.

Stephen Chernicoff's four volume set, *Macintosh Revealed* (Hayden Books), is a step-by-step introduction to Macintosh programming. Chernicoff shows you how to build a working application and points out the parts of *Inside Macintosh* you really need to know as opposed to the parts you just need to be aware of. The programs in the books are written in MPW Pascal, but they can be translated to THINK C or Symantec C++ without great difficulty.

Scott Knaster is the author of two books on Macintosh programming. The first, *How to Write Macintosh Software* (Addison-Wesley), teaches you about the Toolbox and provides some valuable tips about debugging Macintosh programs. The second book, *Macintosh Programming Secrets* (Addison-Wesley), explains the conventions and techniques that have become standard in Macintosh programs. It also contains information about the Macintosh II and the Macintosh SE. These books are more technical than *Macintosh Revealed* and contain pictures, diagrams, and examples.

If you frequently use ResEdit to create and edit resource files, you may want the *ResEdit 2.1 Reference* (Addison-Wesley) by Apple Computer, Inc. It explains how to edit standard resource types and how to extend ResEdit by adding your own resource pickers and editors.

Finally, *MacTech Magazine* is the leading technical journal for Macintosh programming. The articles range from tutorial examples to advanced techniques. *MacTech Magazine* covers several languages, including C, C++, and Pascal. Most examples are written in Symantec C++, THINK C, and THINK Pascal. (All of the programs described in the magazine are available on disk.)

Apple Computer, Inc.

Apple Computer is naturally one of the best places for information about Macintosh programming. Apple administers the Apple Partner

and Apple Associate program for commercial and non-commercial software developers. For more information, contact Apple:

Apple Computer, Inc.
20525 Mariani Avenue, MS 75-2C
Cupertino, CA 95014

(408) 974-4897

Apple Programmer's and Developer's Association (APDA)

The Apple Programmer's and Developer's Association (APDA) is Apple's in-house membership organization that distributes technical information to programmers and developers. APDA is a great source for Technical Notes, programming utilities, reference books, and information about announced (but unreleased) products. For information about membership and products, contact APDA directly:

Apple Programmer's and Developers Association (APDA)
Apple Computer, Inc.
20525 Mariani Avenue, MS 33G
Cupertino, CA 95014-6299

(800) 282-2732 (USA)
(800) 637-0029 (Canada)
(408) 562-3910 (Other)
(408) 562-3971 (Fax)

CompuServe

Symantec has a forum on CompuServe specifically for its customers. Simply type `GO SYMDEVTOOL` at any ! prompt. You'll find discussions here about programming in general and THINK C and THINK Pascal in particular. The data libraries contain utilities as well as sources for many programs.

CompuServe also has an Apple developers forum. Just type `GO MACDEV` at any ! prompt. This forum is a good place to get in touch with the Macintosh programming community.